

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/845,751
Filing Date April 30, 2001
Inventor Bernhard J. Scholz et al.
Group Art Unit 2178
Examiner Paula, Cesar B.
Attorney's Docket No. GE1-004US
Confirmation No. 3458
Title: Automatic Generation of Forms with Input Validation

APPEAL BRIEF

To: Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

From: Allan T. Sponseller (Tel. 509-324-9256 x215; Fax 509-323-8979)
Lee & Hayes, PLLC
421 W. Riverside Ave., Suite 500
Spokane, WA 99201

Pursuant to 37 C.F.R. §41.37, Applicant hereby submits an appeal brief for application 09/845,751, filed April 30, 2001, within the requisite time from the date of filing the Notice of Appeal. Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

<u>Appeal Brief Items</u>	<u>Page</u>
(1) Real Party in Interest	3
(2) Related Appeals and Interferences	3
(3) Status of Claims	3
(4) Status of Amendments	3
(5) Summary of Claimed Subject Matter	4
(6) Grounds of Rejection to be Reviewed on Appeal	6
(7) Argument	6
(8) Appendix of Appealed Claims	19
(9) Appendix of Evidence Submitted	25
(10) Appendix of Related Proceedings	26

(1) Real Party in Interest

The real party in interest is GE Capital Corporation, the assignee of all right, title and interest in and to the subject invention.

(2) Related Appeals and Interferences

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

(3) Status of Claims

Claims 1-6 and 19-43 stand rejected and are pending in this Application. Claims 1-6 and 19-43 are appealed. Claims 1, 2, 4, 6, 19, 23, 28, and 32 were previously amended. Claims 7-18 were previously canceled. Claims 1-6 and 19-43 are set forth in the Appendix of Appealed Claims on page 19.

(4) Status of Amendments

A Final Office Action was issued on August 8, 2005.

A Response to the Final Office Action was filed January 6, 2006. Claim 4 was amended as part of this Response.

An Advisory Action was mailed February 6, 2006, indicating that the Response to the Final Office Action would be entered for purposes of appeal.

Appellant filed a Notice of Appeal on February 8, 2006 in response to the Final Office Action and the Advisory Action.

(5) Summary of Claimed Subject Matter

A concise explanation of each of the independent claims is included in this Summary section, including specific reference characters. These specific reference characters are examples of particular elements of the drawings for certain embodiments of the claimed invention, and the claims are not limited to solely the elements corresponding to these reference characters.

With respect to independent claim 1, as discussed for example at page 30, line 4 through page 39, line 23, and page 58, line 3 through page 59, line 15, one or more computer-readable media comprising computer-executable instructions that, when executed, direct a processor to perform acts including identifying (906) a custom field (802) on a source code form definition (806) and one or more restrictions (804) on an input to the custom field, and identifying validation code (816) that, when executed, validates that the input conforms to the one or more restrictions (804). The acts further include adding, to a new form definition (818) that includes a non-custom field corresponding to the custom field, the identified validation code.

With respect to independent claim 19, as discussed for example at page 30, line 4 through page 39, line 23, and page 58, line 3 through page 59, line 15, a computerized method includes identifying (906), from an input form definition written in a source code (806), one or more desired fields to be included on a form via which data can be input, and automatically adding validation code to source code of the form (910, 914, 916), wherein the validation code is based at least in part on the one or more desired fields (802) and one or more desired input restrictions (804) associated with the one or more desired fields (802).

With respect to independent claim 23, as discussed for example at page 30, line 4 through page 39, line 23, and page 58, line 3 through page 59, line 15, a system includes a form analyzer (810) configured to identify one or more custom tags in a source code form definition (806), and a tag replacement module (812), coupled to the form analyzer (810), configured to replace each of the one or more custom tags with another tag, and further to add, to a form definition (818), for each of the one or more custom tags, validation code to validate subsequent inputs to a field corresponding to the tag.

With respect to independent claim 32, as discussed for example at page 30, line 4 through page 39, line 23, and page 58, line 3 through page 59, line 15, a method includes receiving (902) a form definition (806) including one or more custom tags (802). Each custom tag (802) corresponds to a data input, and each custom tag (802) includes one or more associated input restrictions. The method further includes, for each of the one or more custom tags (802), identifying a replacement non-custom tag, adding (908) the identified replacement non-custom tag to a new form definition (818), identifying validation code (816) that, when executed based on an input corresponding to the tag, validates whether the associated input restrictions are satisfied, and adding the identified validation code to the new form definition (818).

With respect to independent claim 38, as discussed for example at page 30, line 4 through page 39, line 23, and page 58, line 3 through page 59, line 15, a data structure includes a first portion (802) identifying an input field for a form, and a second portion (804) identifying one or more restrictions on inputs to the input

field, and further identifying validation code to be added to a page to enforce the one or more restrictions on inputs to the input field.

(6) Grounds of Rejection to be Reviewed on Appeal

Claims 1, 19, and 43 stand rejected under 35 U.S.C. §102(e) as being unpatentable over U.S. Patent No. 6,292,827 to Raz.

Claims 2-6, 20-24, and 26-42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,292,827 to Raz in view of Laura Lemay's Workshop JavaScript.

Claim 25 stands rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,292,827 to Raz in view of Laura Lemay's Workshop JavaScript and further in view of U.S. Patent No. 6,832,369 to Kryka.

(7) Argument

A. Rejection under 35 U.S.C. §102(e) as being unpatentable over U.S. Patent No. 6,292,827 to Raz.

Claims 1, 19, and 43 stand rejected under 35 U.S.C. §102(e) as being unpatentable over U.S. Patent No. 6,292,827 to Raz (hereinafter "Raz").

Raz is directed to information transfer systems and method with dynamic distribution of data, control and management of information (see, Title). Raz discusses a forms generator/processor that provides the ability to scan paper-based forms and convert them to HTML documents using OCR technology (see, col. 12, lines 37-40). Form blanks are automatically converted to fields that can be typed

into (see, col. 12, lines 40-41). Validation functions can be added to each field to do basic data checking and validation at the client (see, col. 12, lines 44-45).

1. Claim 1

Claim 1 recites:

One or more computer-readable media comprising computer-executable instructions that, when executed, direct a processor to perform acts comprising:

identifying a custom field on a source code form definition and one or more restrictions on an input to the custom field;

identifying validation code that, when executed, validates that the input conforms to the one or more restrictions; and

adding, to a new form definition that includes a non-custom field corresponding to the custom field, the identified validation code.

Appellant respectfully submits that no such identifying and adding is disclosed in Raz.

Raz discusses scanning paper-based forms and converting them to HTML documents using OCR technology (see, col. 12, lines 37-49). Claim 1, on the other hand, recites identifying a custom field on a **source code** form definition and one or more restrictions on an input to the custom field. In order for Raz to disclose the identifying of claim 1, the paper-based forms of Raz would have to be the **source code** form definition recited in claim 1. However, Appellant respectfully submits that a paper-based form is not source code. As such, Appellant respectfully submits that the paper-based forms of Raz cannot disclose the source code form definition of claim 1.

For at least these reasons, Appellant respectfully submits that claim 1 is allowable over Raz.

2. Claim 19

Claim 19 recites:

A computerized method comprising:
identifying, from an input form definition written in a source code, one or more desired fields to be included on a form via which data can be input; and

automatically adding validation code to source code of the form, wherein the validation code is based at least in part on the one or more desired fields and one or more desired input restrictions associated with the one or more desired fields.

Appellant respectfully submits that no such identifying and automatically adding validation code is disclosed in Raz.

Raz discusses scanning paper-based forms and converting them to HTML documents using OCR technology (see, col. 12, lines 37-49). Claim 19, on the other hand, recites identifying, from an input form definition **written in a source code**, one or more desired fields to be included on a form via which data can be input. In order for Raz to disclose the identifying of claim 19, the paper-based forms of Raz would have to be the form definition **written in a source code** recited in claim 19. However, Appellant respectfully submits that a paper-based form is not a form definition written in a source code. As such, Appellant respectfully submits that the paper-based forms of Raz cannot disclose the form definition written in a source code of claim 19.

For at least these reasons, Appellant respectfully submits that claim 19 is allowable over Raz.

3. Claim 43

Claim 43 depends from claim 1 and Appellant respectfully submits that claim 43 is allowable over Raz for at least the reasons discussed above with respect to claim 1. Furthermore, claim 43 recites:

One or more computer-readable media as recited in claim 1, wherein the computer-executable instructions further direct the processor to perform acts comprising:

identifying, on the source code form definition, one or more restrictions for the custom field; and

using, in identifying the validation code, the one or more restrictions.

Appellant respectfully submits that no such identifying and using is disclosed in Raz.

Raz discusses scanning paper-based forms and converting them to HTML documents using OCR technology (see, col. 12, lines 37-49). Claim 43, on the other hand, recites identifying, **on the source code form definition**, one or more restrictions for the custom field. In order for Raz to disclose the identifying of claim 43, there would need to be some disclosure in Raz of identifying, on the paper-based form of Raz, one or more restrictions for a custom field. However, Appellant respectfully submits that nowhere in Raz is there any discussion or mention of identifying **on the paper-based forms of Raz**, any restrictions. Without any such discussion or mention, Appellant respectfully submits that Raz cannot disclose the identifying of claim 43.

For at least these reasons, Appellant respectfully submits that claim 43 is allowable over Raz.

B. Rejection under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,292,827 to Raz in view of Laura Lemay's Workshop JavaScript.

Claims 2-6, 20-24, and 26-42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,292,827 to Raz (hereinafter "Raz") in view of Laura Lemay's Workshop JavaScript (hereinafter "Lemay").

Lemay is directed to validating form data with event handlers (see, p. 132). This validation of Lemay refers to checking each field to ensure that it contains a proper value and advising the user if it is incorrect (see, p. 132).

1. Claims 2-6

Claims 2-6 depend from claim 1, and Appellant respectfully submits that claims 2-6 are allowable over Raz at least because of their dependency on claim 1. Furthermore, the validating of form data with event handlers of Lemay is not cited as curing, and does not cure, the deficiencies of Raz discussed above with respect to claim 1. For at least these reasons, Appellant respectfully submits that claims 2-6 are allowable over Raz in view of Lemay.

2. Claims 20-22

Claims 20-22 depend from claim 19, and Appellant respectfully submits that claims 20-22 are allowable over Raz at least because of their dependency on claim 19. Furthermore, the validating of form data with event handlers of Lemay is not cited as curing, and does not cure, the deficiencies of Raz discussed above

with respect to claim 19. For at least these reasons, Appellant respectfully submits that claims 20-22 are allowable over Raz in view of Lemay.

3. Claims 23, 24 and 26-31

With respect to claim 23, claim 23 recites:

A system comprising:
a form analyzer configured to identify one or more custom tags in a source code form definition; and
a tag replacement module, coupled to the form analyzer, configured to replace each of the one or more custom tags with another tag, and further to add, to a form definition, for each of the one or more custom tags, validation code to validate subsequent inputs to a field corresponding to the tag.

Appellant respectfully submits that no such form analyzer and tag replacement module is disclosed in Raz in view of Lemay.

In the August 8, 2005 Final Office Action, at p. 20, Lemay is cited as disclosing the one or more custom tags in a source code form definition, and replace each of the one or more custom tags with another tag, and a field corresponding to the tag. As discussed above, however, Lemay discusses validating form data with event handlers. Lemay also discusses HTML forms having validation (see, pp. 135-137). However, nowhere in Lemay is there any discussion or mention of **identification of one or more custom tags in a source code form definition**. Without any such discussion or mention, Appellant respectfully submits that Lemay cannot disclose or suggest to identify one or more custom tags in a source code form definition as recited in claim 23. Lemay discusses validating form data with event handlers, and HTML forms having validation, but does not discuss identifying custom tags in a source code form

definition as recited in claim 23. As such, Appellant respectfully submits that Lemay cannot disclose or suggest to identify one or more custom tags in a source code form definition and to replace each of the one or more custom tags with another tag as recited in claim 23.

With respect to Raz, Raz discusses scanning paper-based forms and converting them to HTML documents using OCR technology (see, col. 12, lines 37-49). Claim 23, on the other hand, recites to identify one or more custom tags in a source code form definition. In order for Raz to disclose the identification of claim 23, the paper-based forms of Raz would have to be the **source code** form definition recited in claim 23. However, Appellant respectfully submits that a paper-based form is not a source code form definition. As such, Appellant respectfully submits that the paper-based forms of Raz cannot disclose the source code form definition of claim 23.

For at least these reasons, Appellant respectfully submits that claim 23 is allowable over Raz in view of Lemay.

With respect to claims 24 and 26-31, claims 24 and 26-31 depend from claim 23, and Appellant respectfully submits that claims 24 and 26-31 are allowable over Raz in view of Lemay at least because of their dependency on claim 23. For at least these reasons, Appellant respectfully submits that claims 24 and 26-31 are allowable over Raz in view of Lemay.

4. Claim 32-35 and 37

Claim 32 recites:

A method comprising:
receiving a form definition including one or more custom tags, wherein each custom tag corresponds to a data input, and wherein each custom tag includes one or more associated input restrictions; and
for each of the one or more custom tags,
identifying a replacement non-custom tag,
adding the identified replacement non-custom tag to a new form definition,
identifying validation code that, when executed based on an input corresponding to the tag, validates whether the associated input restrictions are satisfied, and
adding the identified validation code to the new form definition.

Appellant respectfully submits that no receiving a form definition as recited in claim 32 is disclosed in Raz in view of Lemay.

Raz discusses scanning paper-based forms and converting them to HTML documents using OCR technology (see, col. 12, lines 37-49). Claim 32, on the other hand, recites receiving a form definition including one or more custom tags, wherein each custom tag corresponds to a data input, and **wherein each custom tag includes one or more associated input restrictions**. In order for Raz to disclose the receiving of claim 32, the paper-based forms of Raz would need to include one or more custom tags, and each custom tag would need to include one or more associated input restrictions. However, Appellant respectfully submits that nowhere in Raz is there any discussion or mention of the paper-based forms of Raz including custom tags that include input restrictions. Without any such discussion or mention, Appellant respectfully submits that Raz cannot disclose or suggest receiving a form definition including one or more custom tags, wherein

each custom tag corresponds to a data input, and wherein each custom tag includes one or more associated input restrictions as recited in claim 32.

With respect to Lemay, the validating of form data with event handlers of Lemay is not cited as curing, and does not cure, these deficiencies of Raz.

For at least these reasons, Appellant respectfully submits that claim 32 is allowable over Raz in view of Lemay.

With respect to claims 33-35 and 37, claims 33-35 and 37 depend from claim 32, and Appellant respectfully submits that claims 33-35 and 37 are allowable over Raz in view of Lemay at least because of their dependency on claim 32. For at least these reasons, Appellant respectfully submits that claims 33-35 and 37 are allowable over Raz in view of Lemay.

5. Claim 36

Claim 36 depends from claim 32 and Appellant respectfully submits that claim 36 is allowable over Raz in view of Lemay at least because of its dependency on claim 32. Furthermore, claim 36 recites:

A method as recited in claim 32, wherein each input custom tag includes one or more attributes that identify the one or more associated input restrictions, and wherein each of the one or more attributes includes an indication of the attribute and a corresponding value that data input corresponding to the tag is to be restricted to.

Appellant respectfully submits that there is no disclosure or suggestion in Raz or Lemay of each input custom tag including one or more attributes that identify the one or more associated input restrictions, and wherein each of the one or more attributes includes an indication of the attribute and a corresponding value that data input corresponding to the tag is to be restricted to as recited in claim 36.

As discussed above, Raz is directed to scanning paper-based forms and converting them to HTML documents using OCR technology. In order for there to be a disclosure or suggestion of the one or more attributes of claim 36 in the combination of Raz and Lemay, there would need to be some discussion or mention of one or more attributes being included on the paper-based form of Raz. As the one or more attributes in claim 36 include an indication of the attribute and a corresponding value that data input corresponding to the tag is to be restricted to, there would further need to be some discussion or mention in the combination of Raz and Lemay of the indication of the attribute and the corresponding value that data input corresponding to the tag is to be restricted to being included on the paper-based form of Raz. However, nowhere in Raz or Lemay is there any discussion or mention of such being included on the paper-based form of Raz. Without any such discussion or mention, Appellant respectfully submits that Raz cannot disclose or suggest wherein each input custom tag includes one or more attributes that identify the one or more associated input restrictions, and wherein each of the one or more attributes includes an indication of the attribute and a corresponding value that data input corresponding to the tag is to be restricted to as recited in claim 36.

For at least these reasons, Appellant respectfully submits that claim 36 is allowable over Raz in view of Lemay.

6. Claim 38-42

Claim 38 recites:

A data structure comprising:
a first portion identifying an input field for a form; and

a second portion identifying one or more restrictions on inputs to the input field, and further identifying validation code to be added to a page to enforce the one or more restrictions on inputs to the input field.

Appellant respectfully submits that Raz in view of Lemay does not disclose or suggest a data structure having the first and second portions recited in claim 38.

In the August 8, 2005 Final Office Action at pp. 21-22, Raz at col. 12, lines 36-49 is cited as teaching the data structure of claim 38. The HTML form fields of Raz are relied on as disclosing the input field for a form as recited in claim 38. Using the language of claim 38, however, Raz would need to disclose a data structure that includes **both** a first portion identifying those HTML form fields **and** a second portion identifying one or more restrictions on inputs to the HTML form fields and further identifying validation code to be added to a page to enforce the one or more restrictions on inputs to the HTML form fields.

However, Appellant respectfully submits that Raz does not discuss or mention any such data structure that includes **both** such portions. Although Raz states that validation functions can be added to each field to do basic data checking and validation at the client (see, col. 12, lines 44-45), there is no discussion or mention in Raz of where such validation functions come from, much less any discussion or mention that a second portion of the data structure identifying the HTML form fields identifies one or more restrictions on inputs to the HTML form fields and further identifies **validation code to be added** to a page to enforce the restrictions. The mere mention of adding validation functions to fields of an HTML form does not disclose the same data structure identifying **both** the fields and validation code **to be added** to a page to enforce restrictions on inputs to those fields.

With respect to Lemay, the validating of form data with event handlers of Lemay is not cited as curing, and does not cure, these deficiencies of Raz.

For at least these reasons, Appellant respectfully submits that claim 38 is allowable over Raz in view of Lemay.

With respect to claims 39-42, claims 39-42 depend from claim 38, and Appellant respectfully submits that claims 39-42 are allowable over Raz in view of Lemay at least because of their dependency on claim 38. For at least these reasons, Appellant respectfully submits that claims 39-42 are allowable over Raz in view of Lemay.

C. Rejection under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,292,827 to Raz in view of Laura Lemay's Workshop JavaScript and further in view of U.S. Patent No. 6,832,369 to Kryka.

Claim 25 stands rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,292,827 to Raz (hereinafter "Raz") in view of Laura Lemay's Workshop JavaScript (hereinafter "Lemay") and further in view of U.S. Patent No. 6,832,369 to Kryka (hereinafter "Kryka").

1. Claim 25

Kryka is directed to the initialization of static data in object oriented systems (see, Title). In Kryka, a compiler for object-oriented programming code in a language which employs run-time static initialization semantics (such as the JAVA language) analyzes the static initialization code to find a subset of initialization statements which must execute under all conditions, creates an image

of the static storage in which the variables initialized by statements in the subset are pre-initialized at compile time, and removes statements in the subset from the final compiled code of machine level instructions (see, col. 2, lines 44-53).

Claim 25 depends from claim 23 and Appellant respectfully submits that claim 25 is allowable over Raz in view of Lemay at least because of its dependency on claim 23. Furthermore, the initialization of static data in object oriented systems of Kryka is not cited as curing, and does not cure, the deficiencies of Raz in view of Lemay discussed above with respect to claim 23. For at least these reasons, Appellant respectfully submits that claim 25 is allowable over Raz in view of Lemay and Kryka.

Conclusion

The Office's basis and supporting rationale for the § 102(c) and § 103(a) rejections is not supported by the teaching of the cited references. Appellant respectfully requests that the rejections be overturned and that pending claims 1-6 and 19-43 be allowed to issue.

Respectfully Submitted,

Dated: 7/10/86

By: _____

Allan T. Sponseller
Lee & Hayes, PLLC
Reg. No. 38,318
(509) 324-9256 ext. 215

(8) Appendix of Appealed Claims

1. (Previously presented) One or more computer-readable media comprising computer-executable instructions that, when executed, direct a processor to perform acts comprising:

identifying a custom field on a source code form definition and one or more restrictions on an input to the custom field;

identifying validation code that, when executed, validates that the input conforms to the one or more restrictions; and

adding, to a new form definition that includes a non-custom field corresponding to the custom field, the identified validation code.

2. (Previously presented) One or more computer-readable media as recited in claim 1, wherein the computer-executable instructions further direct the processor to perform acts comprising:

adding, to the new form definition, a reference to the identified validation code that, when executed by another processor, causes the other processor to execute the identified validation code.

3. (Original) One or more computer-readable media as recited in claim 1, wherein the identifying validation code comprises identifying pre-defined validation code.

4. (Previously presented) One or more computer-readable media as recited in claim 1, wherein the source code form definition that defines the custom field includes a tag corresponding to the custom field.

5. (Original) One or more computer-readable media as recited in claim 1, wherein the input comprises a user input.

6. (Previously presented) One or more computer-readable media as recited in claim 1, wherein the identifying validation code comprises:

identifying a custom tag corresponding to the custom field, wherein the custom tag includes an indication of one or more attributes, and wherein each of the one or more attributes includes a value indicating what input corresponding to the non-custom field is to be restricted to; and

identifying, from a plurality of pieces of validation code, the validation code corresponding to the one or more attributes of the custom tag.

19. (Previously presented) A computerized method comprising:

identifying, from an input form definition written in a source code, one or more desired fields to be included on a form via which data can be input; and

automatically adding validation code to source code of the form, wherein the validation code is based at least in part on the one or more desired fields and one or more desired input restrictions associated with the one or more desired fields.

20. (Original) A method as recited in claim 19, wherein the identifying comprises identifying a custom tag corresponding to each of the one or more desired fields, wherein each custom tag has one or more validation attributes, and wherein each validation attribute includes an indication of the attribute and a corresponding value that input corresponding to the custom tag is to be restricted to.

21. (Original) A method as recited in claim 19, wherein the input comprises a user input.

22. (Original) A method as recited in claim 19, wherein the automatically adding comprises:

- generating a temporary form definition;
- adding execution code to the temporary form definition;
- executing the execution code to add the validation code to the temporary form definition; and
- outputting, as the source code, the temporary form definition.

23. (Previously presented) A system comprising:
a form analyzer configured to identify one or more custom tags in a source code form definition; and

a tag replacement module, coupled to the form analyzer, configured to replace each of the one or more custom tags with another tag, and further to add, to a form definition, for each of the one or more custom tags, validation code to validate subsequent inputs to a field corresponding to the tag.

24. (Original) A system as recited in claim 23, wherein the inputs comprise user inputs.

25. (Original) A system as recited in claim 23, wherein the system comprises a compiler.

26. (Original) A system as recited in claim 23, wherein each of the other tags with which the tag replacement module replaces a custom tag is a HyperText Markup Language (HTML) tag.

27. (Original) A system as recited in claim 23, wherein the tag replacement module is further configured to add a reference to the added validation code.

28. (Previously presented) A system as recited in claim 23, wherein the tag replacement module is further configured to generate a new document corresponding to the form definition, to replace each of the one or more custom tags with another tag by adding the other tag to the new document, and to add validation code by adding the validation code to the new document.

29. (Original) A system as recited in claim 23, wherein a plurality of the one or more custom tags have restrictions corresponding to the same validation code, and wherein the tag replacement module is further configured to add the same validation code only once.

30. (Original) A system as recited in claim 23, further comprising a tag library, coupled to the tag replacement module, to store the validation code.

31. (Original) A system as recited in claim 30, wherein the tag library is further to store an identification of the one or more custom tags.

32. (Previously presented) A method comprising:
receiving a form definition including one or more custom tags, wherein each custom tag corresponds to a data input, and wherein each custom tag includes one or more associated input restrictions; and
for each of the one or more custom tags,
identifying a replacement non-custom tag,
adding the identified replacement non-custom tag to a new form definition,

identifying validation code that, when executed based on an input corresponding to the tag, validates whether the associated input restrictions are satisfied, and

adding the identified validation code to the new form definition.

33. (Original) A method as recited in claim 32, wherein the method further comprises, for each of the one or more custom tags:

adding, to the new form definition, a reference to invoke the added validation code.

34. (Original) A method as recited in claim 32, wherein the receiving further comprises receiving, as part of the form definition, one or more non-custom tags, and wherein the method further comprises adding each of the non-custom tags to the new form definition.

35. (Original) A method as recited in claim 32, wherein the data input comprises data input by a user.

36. (Original) A method as recited in claim 32, wherein each input custom tag includes one or more attributes that identify the one or more associated input restrictions, and wherein each of the one or more attributes includes an indication of the attribute and a corresponding value that data input corresponding to the tag is to be restricted to.

37. (Original) A method as recited in claim 32, wherein adding the identified validation code comprises:

adding execution code to the new form definition; and

executing the execution code to add the identified validation code to the new form definition.

38. (Original) A data structure comprising:
a first portion identifying an input field for a form; and
a second portion identifying one or more restrictions on inputs to the input field, and further identifying validation code to be added to a page to enforce the one or more restrictions on inputs to the input field.

39. (Original) A data structure as recited in claim 38, wherein the data structure comprises a text markup language document.

40. (Original) A data structure as recited in claim 38, wherein the first portion further identifies a type of the input field.

41. (Original) A data structure as recited in claim 38, wherein the second portion comprises a set of one or more attributes and, for each attribute, an associated value for the attribute.

42. (Original) A data structure as recited in claim 38, wherein the input field is for user-input of data.

43. (Previously presented) One or more computer-readable media as recited in claim 1, wherein the computer-executable instructions further direct the processor to perform acts comprising:

identifying, on the source code form definition, one or more restrictions for the custom field; and

using, in identifying the validation code, the one or more restrictions.

(9) Appendix of Evidence Submitted

None.

(10) Appendix of Related Proceedings

None.